

IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1 – 29. (Canceled)

30. (New) A system comprising:
a network; and
a plurality of computing nodes coupled via the network;
wherein the plurality of nodes includes a first node operable to:
 create a first file representing a first version of a data object;
 detect a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object;
 in response to detecting the conflict:
 modify a tree structure representing the data object to reflect the conflict, wherein modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object, wherein the first replica of the first version of the data object represents the second version of the data object and the second replica of the first version of the data object represents the third version of the data object;
 create a second file representing the second version of the data object; and
 create a third file representing the third version of the data object.
31. (New) The system of claim 30,
wherein creating the second file and the third file comprises creating the second file and the third file in a common directory.
32. (New) The system of claim 31,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory with the first file.

33. (New) The system of claim 30,
wherein the first file has a first name;
wherein creating the second file comprises creating the second file with a second name based on the first name; and
wherein creating the third file comprises creating the third file with a third name based on the first name.

34. (New) The system of claim 30,
wherein the first replica of the first version of the data object is stored on a second node and the second replica of the first version of the data object is stored on a third node.

35. (New) The system of claim 30,
wherein the conflict between the two replicas is caused by update operations that update the two replicas.

36. (New) The system of claim 35,
wherein the update operations that update the two replicas comprise concurrent update operations.

37. (New) The system of claim 30,
wherein the conflict between the two replicas is caused by the two replicas being independently updated in different network partitions.

38. (New) A method comprising:
creating a first file representing a first version of a data object;

detecting a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object;

in response to detecting the conflict:

modifying a tree structure representing the data object to reflect the conflict, wherein modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object, wherein the first replica of the first version of the data object represents the second version of the data object and the second replica of the first version of the data object represents the third version of the data object;

creating a second file representing the second version of the data object; and

creating a third file representing the third version of the data object.

39. (New) The method of claim 38,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory.

40. (New) The method of claim 39,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory with the first file.

41. (New) The method of claim 38,

wherein the first file has a first name;

wherein creating the second file comprises creating the second file with a second name based on the first name; and

wherein creating the third file comprises creating the third file with a third name based on the first name.

42. (New) The method of claim 38,

wherein the first replica of the first version of the data object is stored on a first node and the second replica of the first version of the data object is stored on a second node.

43. (New) A computer-readable memory medium comprising program instructions executable to:

create a first file representing a first version of a data object;

detect a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object;

in response to detecting the conflict:

modify a tree structure representing the data object to reflect the conflict, wherein modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object, wherein the first replica of the first version of the data object represents the second version of the data object and the second replica of the first version of the data object represents the third version of the data object;

create a second file representing the second version of the data object; and

create a third file representing the third version of the data object.

44. (New) The computer-readable memory medium of claim 43, wherein creating the second file and the third file comprises creating the second file and the third file in a common directory.

45. (New) The computer-readable memory medium of claim 44, wherein creating the second file and the third file comprises creating the second file and the third file in a common directory with the first file.

46. (New) The computer-readable memory medium of claim 43,

wherein the first file has a first name;

wherein creating the second file comprises creating the second file with a second name based on the first name; and

wherein creating the third file comprises creating the third file with a third name based on the first name.

47. (New) The computer-readable memory medium of claim 43,
wherein the first replica of the first version of the data object is stored on a first node and the second replica of the first version of the data object is stored on a second node.

48. (New) A computing node comprising:
one or more processors; and
a memory storing program instructions;
wherein the one or more processors are operable to execute the program instructions to:

create a first file representing a first version of a data object;
detect a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object;

in response to detecting the conflict:
modify a tree structure representing the data object to reflect the conflict, wherein modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object, wherein the first replica of the first version of the data object represents the second version of the data object and the second replica of the first version of the data object represents the third version of the data object;

create a second file representing the second version of the data object; and

create a third file representing the third version of the data object.

49. (New) The computing node of claim 48,
wherein creating the second file and the third file comprises creating the second
file and the third file in a common directory.

50. (New) The computing node of claim 49,
wherein creating the second file and the third file comprises creating the second
file and the third file in a common directory with the first file.

51. (New) The computing node of claim 48,
wherein the first file has a first name;
wherein creating the second file comprises creating the second file with a second
name based on the first name; and
wherein creating the third file comprises creating the third file with a third name
based on the first name.